

# 编程能力摸底试题

## A. 斗牛

给定五个 0~9 范围内的整数  $a_1, a_2, a_3, a_4, a_5$ 。如果能从五个整数中选出三个并且这三个整数的和为 10 的倍数（包括 0），那么这五个整数的权值即为剩下两个没被选出来的整数的和对 10 取余的结果，显然如果有多个三元组满足和是 10 的倍数，剩下两个数之和对 10 取余的结果都是相同的；如果选不出这样三个整数，则这五个整数的权值为 -1。

现在给定  $T$  组数据，每组数据包含五个 0~9 范围内的整数，分别求这  $T$  组数据中五个整数的权值。

### 【输入格式】

第一行一个整数  $T$  ( $1 \leq T \leq 1000$ )，表示数据组数。

接下来  $T$  行，每行 5 个 0~9 的整数，表示一组数据。

### 【输出格式】

输出  $T$  行，每行一个整数，表示每组数据中五个整数的权值。

### 【样例输入】

```
4
1 0 0 1 0
1 0 0 8 6
3 4 5 6 7
4 5 6 7 8
```

### 【样例输出】

```
2
-1
-1
0
```

### 【解释】

在第一组 (1 0 0 1 0) 中, 三元组 0 0 0 的和为 0, 是 10 的倍数, 剩余的 1 1 之和为 2, 对 10 取余为 2。

在第二组中, 不存在任何一个三元组只和为 10 的倍数。

在第四组中, 三元组 5 7 8 的和为 20, 是 10 的倍数, 剩余的 4 6 只和为 10, 对 10 取余为 0。

在第五组中, 三元组 0 3 7 和三元组 0 4 6 的和都是 10, 是 10 的倍数, 但是根据简单的数论可知, 如果存在多个三元组满足情况, 那么剩余数字的结果之和对 10 取余是相等的, 在本例中和为 10, 对 10 取余为 0。

## B. 打地鼠

给定  $n$  个整数  $a_1, a_2, \dots, a_n$  和一个  $d$ , 你需要选出若干个整数, 使得将这些整数从小到大排好序之后, 任意两个相邻的数之差都不小于给定的  $d$ , 问最多能选多少个数出来。

### 【输入格式】

第一行两个整数  $n, d$  ( $1 \leq n \leq 10^5, 0 \leq d \leq 10^9$ ), 分别表示整数个数和相邻整数差的下界。

第二行  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9, 1 \leq i \leq n$ ), 表示给定的  $n$  个整数。

### 【输出格式】

仅一行一个整数, 表示答案。

### 【样例输入】

```
6 2
1 4 2 8 5 7
```

### 【样例输出】

```
3
```

### 【解释】

注意, 选出的数在排序后, 相邻两数之差不小于给定值。

比如，对于给定值 2，[1 4 7] 是一个满足条件的选择方案，但是[1 4 5] 却不是，因为  $5 - 4 = 1 < 2$ 。  
在本样例中，[1 4 7]，[1 4 8]，[1 5 7]，[1 5 8]，[2 4 7]，[2 4 8] 都是满足要求的方案，但是无论如何都没有办法得到一个选出 4 个数且满足条件的方案，所以本样例的答案为 3。

## C. 排队打饭

下课了，有  $n$  位同学陆续赶到食堂进行排队打饭，其中第  $i$  位同学的到达时间为  $a_i$ ，打饭耗时为  $t_i$ ，等待时间上限为  $b_i$ ，即如果其在第  $a_i + b_i$  秒的时刻仍然没有轮到他开始打饭，那么他将离开打饭队列另寻吃饭的地方。问每位同学的开始打饭时间，或者指出其提前离开了队伍（如果这样则输出 -1）。

### 【输入格式】

第一行一个整数  $n$  ( $1 \leq n \leq 10^5$ )，表示来打饭的同学数量。

接下来  $n$  行，每行三个整数  $a_i, t_i, b_i$  ( $1 \leq a_i, t_i, b_i \leq 10^9, 1 \leq i \leq n$ )，分别表示每位同学的到达时间、打饭耗时、等待时间上限。

保证  $a_1 < a_2 < \dots < a_n$

### 【输出格式】

一行  $n$  个整数，表示每位同学的开始打饭时间或者 -1（如果该同学提前离开了队伍）。

### 【样例输入】

```
4
1 3 3
2 2 2
3 9 1
4 3 2
```

### 【样例输出】

```
1 4 -1 6
```

### 【解释】

第一个同学在 1 时刻到达队列，需要 3 个单位时间才能打好饭（也就是说如果在 1 时刻开始打饭，那么将在  $1 + 3 = 4$  时刻打好饭离开），最长等待时间为 3 个单位时间（也就说如果在到达队列之后的 3 单位时间后还没开始给他打饭，他就忍耐不了离开了）。

在本样例中，

1 时刻：第一个同学在 1 时刻到达队列，同时开始了打饭操作（对应输出的第一个值为 1）。

2 时刻：在 2 时刻，第二个同学加入了队列，给第二个同学打饭需要 2 个单位时间，但是如果在等待了 2 个单位时间没给第二个同学打饭的话，第二个同学将离开。

3 时刻：在 3 时刻，第三个同学加入了队列，给第三个同学打饭需要 9 个单位时间，但是如果在等待了 1 个单位时间没给第三个同学打饭的话，第三个同学将离开，换句话说，如果在  $3$ （到达时刻）+ 1（可等待时间长度）= 4 时刻还没给第三个同学打饭，那么第三个同学将离开。

4 时刻：第一个同学在时刻 4 打完饭离开，同时队列里的第二个同学开始打饭（对应输出的第二个值为 4），此时第三个同学没有达到饭，所以第三个同学就在时刻 4 离开了队伍（对应输出的第三个值为 -1）。同时，在时刻 4，第四个同学也加入了队列，第四个同学最长等待到  $4$ （到达时刻）+ 2（可等待时间长度）= 6 时刻。

5 时刻：5 时刻还在给第二个同学打饭，第四个同学还在队列里面排队。

6 时刻：6 时刻，第二个同学打饭完成，同时第四个同学开始打饭（对应输出的第四个值为 6）。

根据上面描述的过程，输出为 1 4 -1 6。

## D. 二叉搜索树

给定一个  $1 \sim n$  的排列  $P$ ，即长度为  $n$ ，且  $1 \sim n$  中所有数字都恰好出现一次的序列。现在按顺序将排列中的元素一一插入到初始为空的二叉搜索树中（左小右大），问最后每个节点的父亲节点的元素是什么。特别地，根节点的父亲节点元素视为 0。

### 【输入格式】

第一行一个整数  $n$  ( $1 \leq n \leq 10^5$ )，表示排列  $P$  中的元素个数。

第二行  $n$  个整数  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq n, 1 \leq i \leq n$ )，表示给定的排列。

### 【输出格式】

一行  $n$  个整数，其中第  $i$  个整数  $a_i$  表示元素  $i$  对应节点的父亲节点的元素。特别地，根节点的父亲节点元素视为 0。

【样例输入】

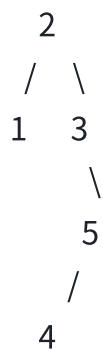
5  
2 3 5 1 4

【样例输出】

2 0 2 5 3

【样例解释】

最后建出来的二叉搜索树如下：



1 的父亲为 2，2 为根结点，所以父亲为 0，3 的父亲为 2，4 的父亲为 5，5 的父亲为 3。

## E.序列

给定一个长为  $n$  的序列  $A$ ，其中序列中的元素都是  $0\sim 9$  之间的整数，对于一个长度同样为  $n$  整数序列  $B$ ，定义其权值为  $|A_i-B_i|$  ( $1\leq i\leq n$ ) 之和加上  $(B_j-B_{j+1})^2$  ( $1\leq j<n$ ) 之和。求所有长为  $n$  的整数序列中，权值最小的序列的权值是多少。

【输入格式】

第一行一个整数  $n$  ( $1\leq n\leq 10^5$ )，表示序列  $A$  的长度。

第二行  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $0\leq a_i\leq 9, 1\leq i\leq n$ )，表示序列  $A$  中的元素。

【输出格式】

仅一行一个整数，表示答案。

【样例输入】

6  
1 4 2 8 5 7

【样例输出】

11

【解释】

A 数组是 [1 4 2 8 5 7]

B 数组可以是 [3 4 4 5 5 6]。

权值为  $|A_i - B_i|$  ( $1 \leq i \leq n$ ) 之和加上  $(B_j - B_{j+1})^2$  ( $1 \leq j < n$ ) 之和。

权值第一部分  $|A_i - B_i|$  ( $1 \leq i \leq n$ ) 之和为：

$$|1 - 3| + |4 - 4| + |2 - 4| + |8 - 5| + |5 - 5| + |7 - 6| = 2 + 0 + 2 + 3 + 0 + 1 = 8$$

权值第二部分  $(B_j - B_{j+1})^2$  ( $1 \leq j < n$ ) 之和为：

$$(3 - 4)^2 + (4 - 4)^2 + (4 - 5)^2 + (5 - 5)^2 + (5 - 6)^2 = 1 + 0 + 1 + 0 + 1 = 3$$

所以总权值为  $8 + 3 = 11$ 。